TALKING GLOVES

Mitali Madhusmita
Atul Agrawal

Department of Electronics and Communication Engineering

National Institute of Technology
Rourkela-769008

May 2015

Talking Gloves

A Thesis submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

By

Mitali Madhusmita

Roll No.: 111EC0178

Atul Agrawal

Roll No.: 111EI0594

Under the Guidance of

Prof. Manish Okade

Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela-769008 (ODISHA)

May 2013

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGG,

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA-769 008

ODISHA, INDIA

CERTIFICATE

This is to certify that the thesis entitled "**Talking Gloves**", submitted to the National Institute of Technology, Rourkela by **Mitali Madhusmita, Roll No**. **111EC0178** and **Atul Agrawal, Roll No**. **111EI0594** for the award of the degree of **Bachelor of Technology** in Department of Electronics and Communication Engineering, is a bonafide work carried out by them under my supervision and guidance.

The candidates have fulfilled all the prescribed requirements. The thesis is based on candidate's own work, is not submitted elsewhere for the award of degree/diploma.

In my opinion, the thesis is in standard fulfilling all the requirements for the award of the degree of **Bachelor of Technology** in Electronics and Communication Engineering.

Prof. Manish Okade
Supervisor

Department of Electronics and Communication Engineering

National Institute of Technology-Rourkela,

Odisha– 769008 (INDIA

ACKNOWLEDGEMENT

We would like to convey our deepest gratitude towards our supervisor, Professor Manish Okade for his support and supervision, and for the valuable knowledge that he shared with us.

We would like to thank our friends and juniors who have helped us to complete the thesis work successfully.

We would like to convey appreciation to our family members, for their encouragement and support.
We thank God for being on our side.

Mitali Madhusmita (111EC0178)
Atul Agrawal (111EI0594)

Table of Contents

LIST OF FIGURES……..i

LIST OF TABLES……...ii

ABSTRACT……..iii

Chapter 1……...1

1.1 Introduction……... 2
1.2 Literature Review……....3
1.3 Motivation……..5
1.4 Objective of our Thesis……....6
1.5 Work done and Thesis organization……6

Chapter 2…….8

System Description

2.1 Introduction …….....9
2.2 Sign Language…….....9
2.2.1 Origin of sign language.....10
2.2.2 Phonology.....11
2.3 System Architecture and Implementation……..11
2.3.1 Flex Sensor.....12
2.3.2 Accelerometer.....13
2.3.3 Arduino Board.....14
2.3.4 Bluetooth Module.....15

Chapter 3..... 17
Result Discussion …….. 17
3.1 Result…….....18
3.1.1 Flex Sensor.....18
3.1.2 Final circuit.....18
3.2 Discussion ……. 27

Chapter 4

Future scope and conclusion…….28

REFERENCES **30**

LIST OF FIGURES

Figure No.	Title	Page No.
Fig1. 1	Flow diagram of work done	6
Fig 2.1	Transmission Circuit	11
Fig 2.2	Reception Circuit	12
Fig 2.3	Flex Sensor	13
Fig 3.1:	Transmission circuit using flex sensor	18
Fig 3.2	Audio Amplifier	24
Fig 3.3	Text to speech conversion circuit	24

LIST OF TABLES

Table No.	Title	Page No.
Table 2.1	Comparison of various Arduino boards	14
Table 2.2:	Hardware Specification of Bluetooth Module HC-05	15
Table 2.3	Software Specification of Bluetooth Module HC-05	16

ABSTRACT

Being an essential part of individual's life and organization's normal functioning, communication plays an integral part of development. Without communication growth of a person is hampered. But everyone is not gifted with the ability to hear and speak. They use sign language to communicate. It is difficult for most of the people who are not familiar with this sign language to communicate without an interpreter. Thus, a system that transcribes symbols in sign languages into plain text can help with real-time communication .Talking gloves is an human interface device which converts mechanism of hand sign language into alphanumerical characters. It is a product in concurrence with assistive engineering to help dumb people of society and improve communication capabilities.

C H A P T E R - 1
I N T R O D U C T I O N

1.1 Introduction

Assistive technology, it is an umbrella term which includes rehabilitative and adaptive devices for people with disabilities and also includes the process used in locating and using them. It promotes independence by enabling differentially abled persons to perform tasks that they were formerly unable to accomplish, or had great difficulty in accomplishing, by providing various methods for interacting with the innovative technology needed to accomplish such tasks. It points to any piece of equipment, or product system, which is acquired commercialisation, modification, or customization and this is used to maintain, or improve functional capabilities of differentially abled individuals.

Sign language is a language which is use body movements to convey the meaning, as opposed to acoustically conveyed sound patterns. It is used by deaf and dumb to express their emotions before others a non-verbal communication delivering some message is known as a gesture.

Talking Gloves is an example of assistive technology that converts sign language into words. A simple glove with accelerometers fitted in for each of the finger, is used for getting acceleration of each finger with respect to a particular hand gesture .Data when fed into Arduino gives PWM signals .These signals are transmitted and received with the help of Bluetooth modules including internal Bluetooth module.. In receiving circuit gesture is recognized and signal (text information) is identified. That text information is converting the speech signal with the help of LM-386 and the words are played with the help of speaker.

1.2 Literature Review:

1.2.1 Flex Sensors

Flex sensor or Bend Sensor is Analog resistor which is work as variable Analog voltage divider. Flex sensors patented technology is depends on resistive carbon thick elements. In the flex sensor there are carbon particles which are moved by bending the sensor. If we bend the sensor then carbon particles will be moved and according to their movement the resistance of flex sensor will be changed and voltage output also changed accordingly. The sensor is connected to three pin device connector in which one pin is ground second is live and third one is output pin. This device will activate the sensor which was is in sleep mode, and enabling them to power down when that is not is in use and when that is greatly decrease the power consumption. It is change the resistance only in one direction. If we increase the bend angle means if we increase the bending movement then resistance will be increased and voltage will be decreased. Input voltage of flex sensor is 5 volt and upcoming output voltage will be 0-5 volt.

1.2.2 Accelerometer

Accelerometer is an electronic device which is used in measuring the tilt and motion. Accelerometer is very useful or says capable for detecting rotation and motion gestures like a shaking or swinging. For making an accelerometer there are so many different ways which we can use for making it. An accelerometer which are used for the piezoelectric effect, they are using microscopic crystal structures. Microscopic crystal structures get stressed by accelerative forces, which is caused a voltage to be generated. A different way to do

piezoelectric effect is by sensing change in capacitance. We have certain capacitance between two microstructure, if both the microstructure are next to each other. The capacitance will be changed, if an accelerative force moves one of the structures. For converting the capacitance to voltage we need to add some circuitry, and after that we will get an accelerometer. There are so many different methods also available including use of hot air bubbles, piezoelectric effect and light.

1.2.3 Arduino

Arduino is a movement, not a microcontroller or not a chip(IC), not only a board (PCB), not a company or manufacture, not only a programing language or not only a computer architecture. Arduino is an open source hardware platform and also an open source development environment which is very easy to learn libraries and language (Arduino is based on a wiring language). Arduino has integrated development environment which is based on processing programing environment. Arduino is available for maximum operating systems means Arduino is available for windows, Linux, mac etc.

Arduino is a tool which is used for making computer which can control or sense more of the physical world than our computer. Based on a simple microcontroller board, it is an open source physical computing platform. For writing software for board, there is development environment.

It is an open source electronics prototyping platform which is based on easy to use, flexible hardware and software. Arduino is intended for designers, artists, hobbyists, and anyone interested in creating an interactive environment or objects.

Arduino is a microcontroller on a circuit board which makes it easy to receive inputs and drives the outputs. A microcontroller is an integrated computer on a chip.

1.2.4 Bluetooth Module HC-05

We are using HC-05 Bluetooth module for our project which is simple to use for Bluetooth Serial Port Protocol (SPP) module and for the transparent wireless serial connection setup, we are designed a Bluetooth HC-05 module. With Serial Port Profile, HC-05 Bluetooth module is a class-2 module, which is configure as either slave or master. For wired serial connection, there is possibility of a drop-in replacement. To establish the connection, we can use HC-05 as a serial port replacement, between MCU, PC to our embedded project. Bluetooth module HC-05 will be simplified our overall design. HC-05 module has the footprint which is too small (12.7mm*27mm). With CMOS technology and Adaptive Frequency Hopping Feature (AFH), HC-05 module will be used CSR Bluecore04-External single chip Bluetooth system.

1.3 Motivation

Communication, being the oxygen of life without which growth of individual is hampered. One is bound to be suppressed if one cannot express. Human beings have been gifted, by nature, with voice that allows them to interact and communicate with each other. Hence, speaking becomes one of the main attributes of humans. Unfortunately, not everybody possesses this capability i.e. hearing *(Daniel Capilla)*. In India, there are around 15 million deaf people (ref. www.def.org). Sign language is the basic alternative communication method between deaf people and several dictionaries of words or single letters have been defined to

make this communication possible .Dumb are a part of our society and they should not be deprived of their rights .Therefore there is a need for development of one such device which simplifies their communication.

1.4 Objective of our thesis

- To develop a system that recognizes hand gesture which distinguishes between four different Hand gestures signifying four numbers in American Sign Language.
- Changing these numbers into words

1.5 Work done and Thesis Organization

1.5.1 Work done

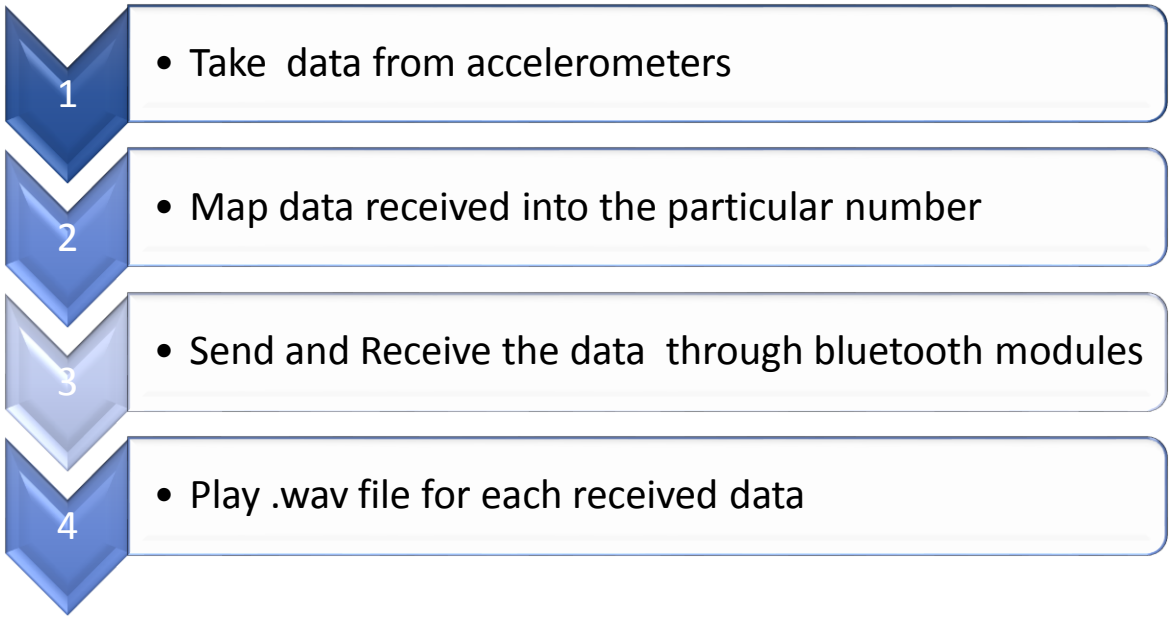


Fig1. 1: Flow diagram of work done

- Take acceleration of each finger as measured by accelerometer and fed into Arduino
- Map this Data into the number which the hand sign meant

- The data is send and received through Bluetooth modules taking internal Bluetooth as relay
- The .wav file is played for the respective number received

1.5.2 Thesis Organization

Chapter 1:

Introduction, Literature Review, Motivation, Work done and Thesis organization.

Chapter 2:

System description and Description of different steps carried out in our project.

Chapter 3:

Simulation Results and Discussion.

Chapter 4:

Future Scope and Conclusion.

C H A P T E R – 2
SYSTEM DESCRIPTION

2.1 Introduction

Talking gloves is a human interface device which converts mechanism of hand sign language into alphanumerical characters. It is a product in concurrence with assistive engineering to help dumb people of society and improve communication capabilities. We are proposing a product in concurrence with assistive engineering to help the differentially- abled. This product targets the deaf and dumb. The objective of this project is to develop such a human interface device which converts the mechanism of American Sign Language into alphanumerical characters, then assist users to show and to communicate with others through voice. The goal of this project is to identify 4 numbers of American Sign Language, then play it through speaker or listen through ear cord

. 2.2 Sign Language

Nowadays, we can find a wide number of sign languages all over the world and almost every spoken language has its respective sign language. American Sign Language (ASL), Algerian Sign Language (ASL), Kenyan Sign Language (KSL), Mauritian Sign Language
(MSL), Irish Sign Language (IRSL), Tanzanian sign language, British Sign Language (BSL), Australian Sign Language (Auslan), German Sign Language (DGS), Indian Sign Language (ISL), and Spanish Sign Language (LSE), Yoruba Sign Language are just a few of them. Among all this large list, American Sign Language is cur-reentry the predominant of any sign language and its grammar has been successfully applied for various other sign languages such as British Sign Language. BSL is not closely related to ASL.

The goal is to provide the reader with a basic knowledge about the sign languages used by deaf and dumb people of the society. The following section will attempt to give a general description of the shared

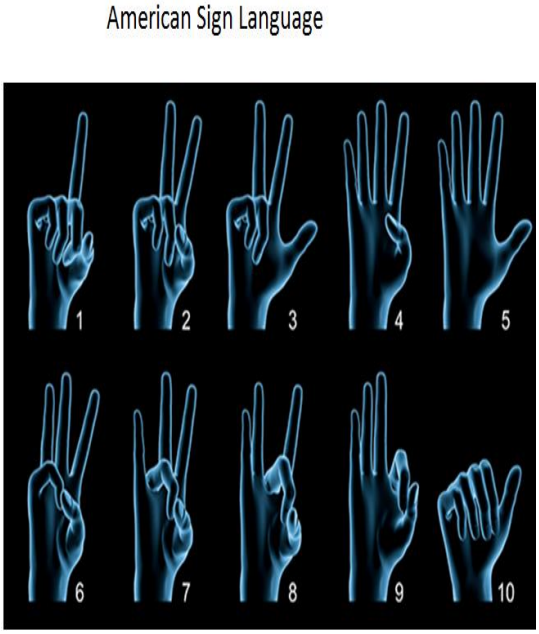
Characteristics among the different sign languages: origin, phonology, and syntax (for the last two, contains an easy-to-understand description). By doing so, people who are not familiar with them will realize how complex it would be to design a whole Sign Language Translator and why the decision to simplify the system without taking into account these characteristics was made in the version of the system introduced here.

2.2.1 Origin of sign language

One of the earliest written records of a sign language is from the fifth century BC, in Plato's *Cratylus*, where Socrates says: "If we hadn't a voice or a tongue, and wanted to express things to one another, wouldn't we try to make signs by moving our hands, head, and the rest of our body, just as dumb people do at present?"

Until the 19th century, most of us know about historical sign that they are limited to the manual alphabets and they were invented to facilitate transfer of words from a spoken to a signed language. Sign language is mainly taught to deaf people, but its origin dates from the beginning history. In fact, gestures are the basic way that kids have to express their feelings until they learn their respective mother tongue.

The starting real study of sign languages is relatively younger compared to spoken languages. It dates from 1960, but today there is not an exact definition of their grammar. Since time immemorial, there is quantitative and qualitative advancement of sign language linguistics, but there are still few problems like the definition of a tool to transcript any sort of sign language.



2.2.2 Phonology

In spoken language, the phonology denotes the study of physical sounds present in human speech (called phonemes). Similarly, the phonology of sign language can be defined. Instead of sounds, the phonemes are considered as the different signs present in a row of hand signs.

They are analyzed taking into account the following characteristics:

- Configuration: Hand shape while doing the sign.
- Orientation of the hand: Where the palm is pointing.
- Non-manual components: Refers to the information provided by the body (facial expression, movements of the shoulders or lip movements).

2.3 System Architecture and Implementation

In our project, we have five accelerometers fitted in each finger. The hand gesture for a particular number provides the required data (i.e. acceleration measured by accelerometer). This data is fed into Arduino Mega 2560. Here the data from all the five fingers is mapped into the number represented by the particular hand gesture. This number is then send from

Bluetooth device hc-05 to internal Bluetooth i.e. hc-06.This internal Bluetooth acts as a relay to connect two external Bluetooth devices, the other one to receive the data.

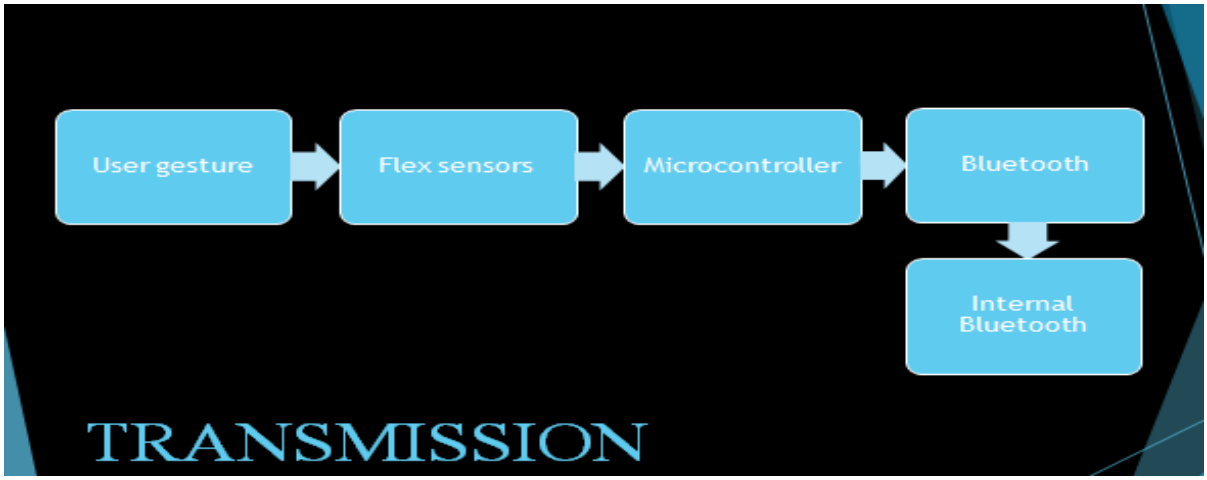


Fig 2.1.transmission Circuit

The received data is passed in to Arduino mega.In Receiver circuit we use Arduino for playing .wav file from Arduino. The .wav file played from SD Card module is amplified with the help of LM-386. Receiver circuit uses a speaker of 8 ohm. SD card inserted should be less than 2GB. Potentiometer and first order filtering circuit is being used for high amplitude and lowering noise respectively.

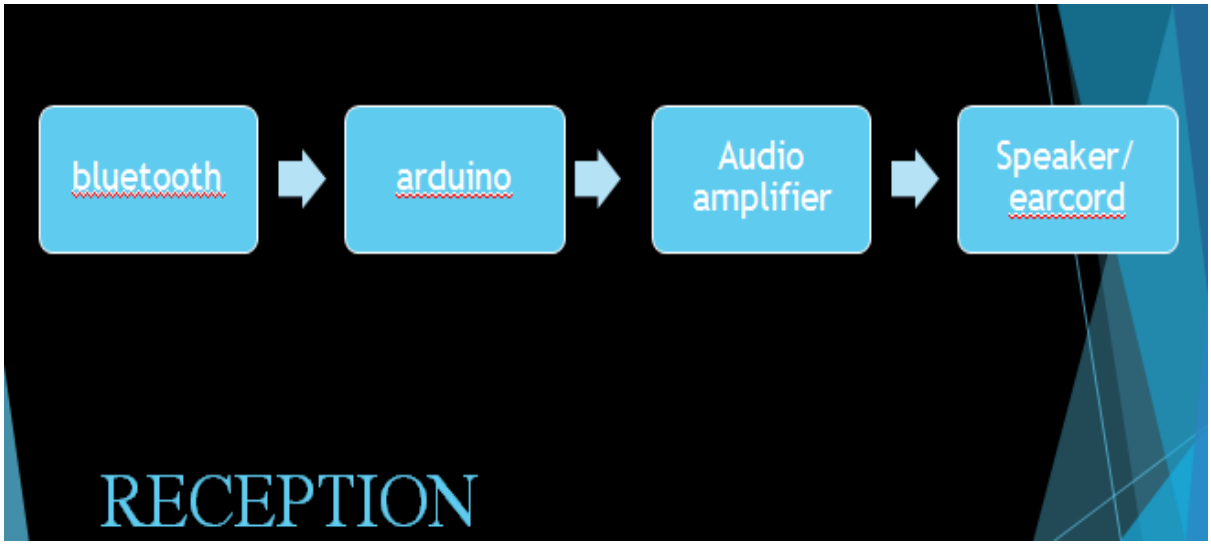


Fig 2.2. Reception Circuit

2.3.1 Flex Sensors

These are based on carbon thick elements. When the thin substrate is bent, the sensor is produced a resistance output which is correlated to the bend radius. Smaller the radius, higher the resistance. These sensors require 5v input and output between 0-5V. The resistivity varies with sensor's degree of bend and the voltage output changing accordingly.

Flex Sensors changes resistance in one direction only. An unflexed sensor has resistance of about 10Kohm. When the flex sensor is bent more, the resistance increases to 30-40Kohm at 90degree. One side of the sensor is printed with a polymer link that has conductive particles embedded in it. When sensor is straight, the particles give the link a resistance of say 30kohm. When sensor is bend away from link, the conductive particles move further apart increasing this resistance(to about 50Kohm).When the sensor straightens out again , the resistance returns to the original value. By measuring the resistance, we can determine how much sensor is bent.

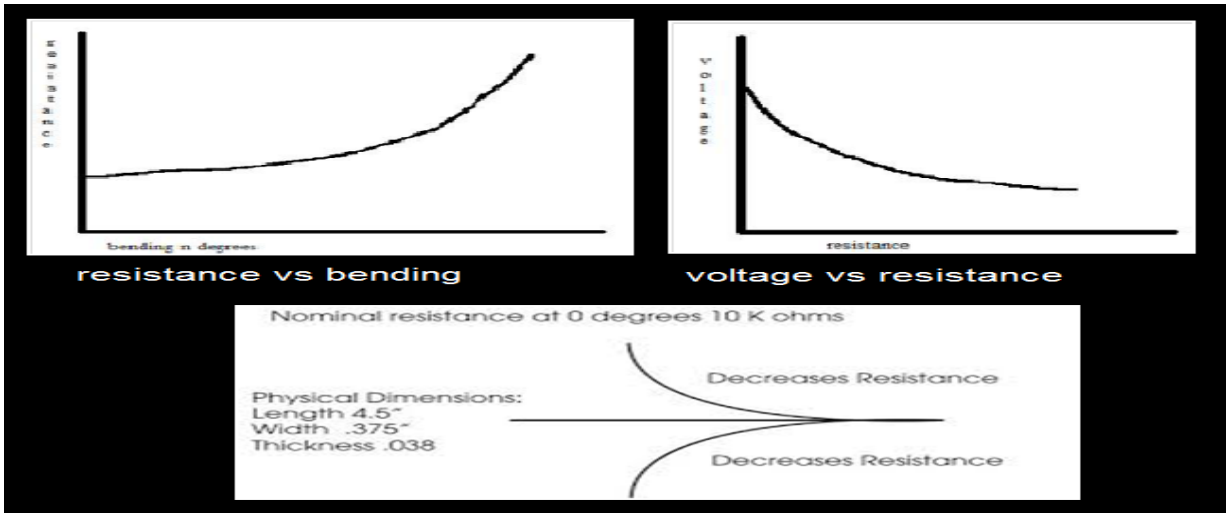


Fig 2.3 Flex Sensor

2.3.2 Accelerometers

An accelerometer is a device that measures proper acceleration. Coordinate acceleration (rate of change) and proper acceleration are not same. Each of the accelerometers provide the x, y and z coordinates acceleration i.e. a change in gravitational force**.**

Accelerometer is an electromechanical device which is used to measure acceleration force. There are two types of Acceleration force for accelerometer, which are: static force, like the constant force of gravity pulling at our feet, and dynamic force – caused by moving or vibrating the accelerometer. Accelerometer is also used for measuring the tilt and motion. It means we also told that accelerometer is an electronic device which are used in measuring the tilt and motion. Accelerometer is capable for detecting rotation and motion gestures like a shaking or swinging.

2.3.3 Arduino Board

After getting analog signal from the flex sensor we need to convert this analog signal to digital signal by ADC. In our project we have used Arduino Mega 2560.Arduino is an open source physical computing platform that is based on a simple microcontroller board and a development environment for writing software. It can be used to develop interactive objects, taking inputs from a variety of lights, etc.

Arduino Board	Family	Clock	UART	Digital	Analog	SRAM	FLASH	EEPROM
UNO	ATMEGA 328	16MHz	1	14	6	2k	22k	1kB
Mega 2560	ATMEGA 2560	16MHz	4	54	16	8k	256k	1kB
Duemilanove (328)	ATMEGA 328	16MHz	1	14	6	2k	22k	1kB
Arduino mini	ATMEGA 328	16MHz	1	14	8	2k	22k	1kB
Arduino PRO 5V	ATMEGA 328P	16MHz	1	14	6	2k	22k	1kB

Comparision of various Arduino boards

Table 2.1: comparison of various Arduino boards

2.3.4 Bluetooth Module

We are using HC-05 Bluetooth module for our project which is simple to use for Bluetooth Serial Port Protocol (SPP) module and for the transparent wireless serial connection setup, we are designed a Bluetooth HC-05 module. With Serial Port Profile, HC-05 Bluetooth module is a class-2 module, which is configure as either slave or master. For wired serial connection, there is possibility of a drop-in replacement. To establish the connection, we can use HC-05 as a serial port replacement, between MCU, PC to our embedded project. Bluetooth module HC-05 will be simplified our overall design. HC-05 module has the footprint which is too small (12.7mm*27mm). With CMOS technology and Adaptive Frequency Hopping Feature (AFH), HC-05 module will be used CSR Bluecore04-External single chip Bluetooth system.

Specification:

There is two types of specification for Bluetooth module HC-05.

Hardware Specification:

Specifications	Range
Sensitivity	80dbm
RF Transmit Power	up to +4dbm
Operation power	1.8V
Frequency range	2.4GHz
.Emission power	4dbm
Speed Range	160kbps to 2.1 Mbps
Power supply	+3.3V, 50 mA

Table 2.2: Hardware Specification of Bluetooth Module HC-05

Software Specification:

Specifications	Range
Default Baud Rate	38400bps
Supported baud rate	9600, 19200, 38400, 57600, 115200, 230400, 460800bps
Data Bits	8
Stop Bit	1
Parity Bit	No
Default auto-pairing pincode	0000

Table 2.3: Software Specification of Bluetooth Module HC-05

C H A P T E R – 3
RESULTS AND DISSCUSSION

3.1 Results

3.1.1 Flex sensors

Single flex sensor was used to know its working properly. The voltage variation after being converted to a digital number was transmitted and received.

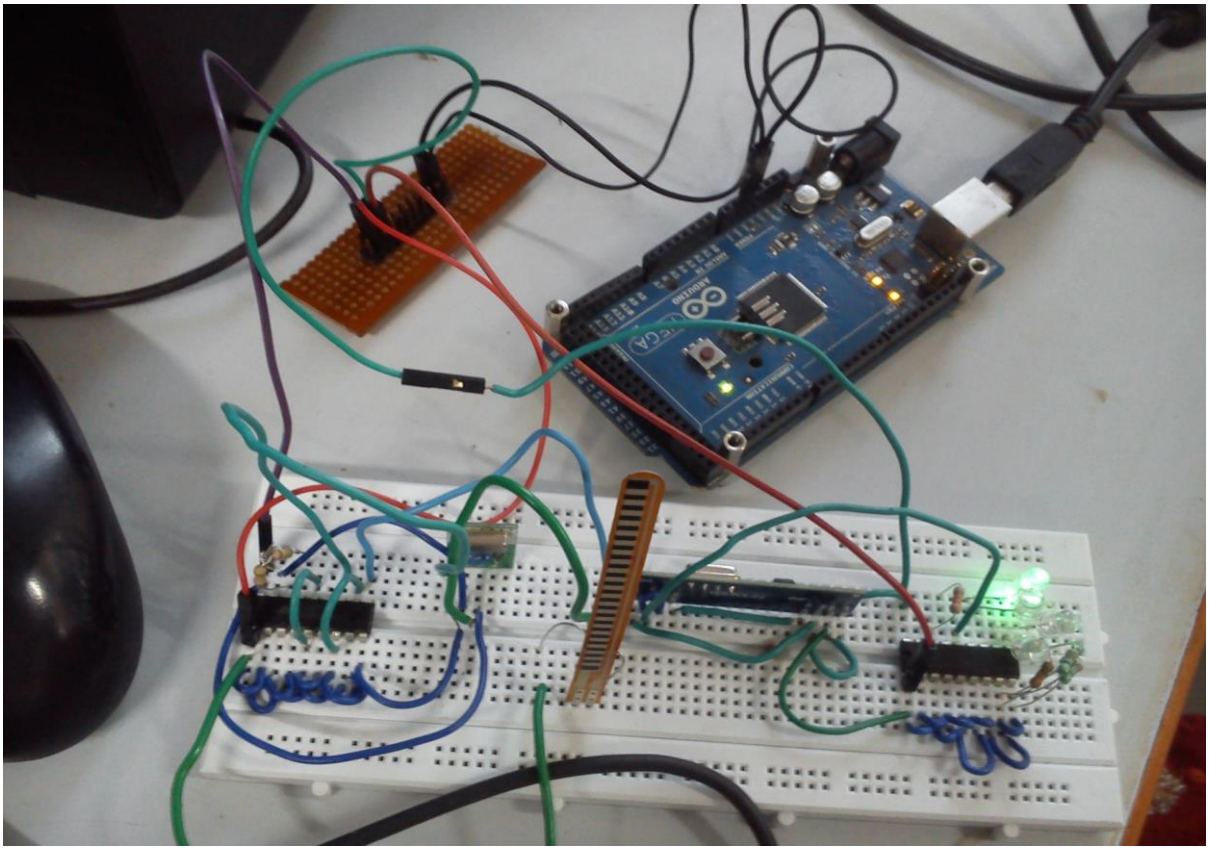
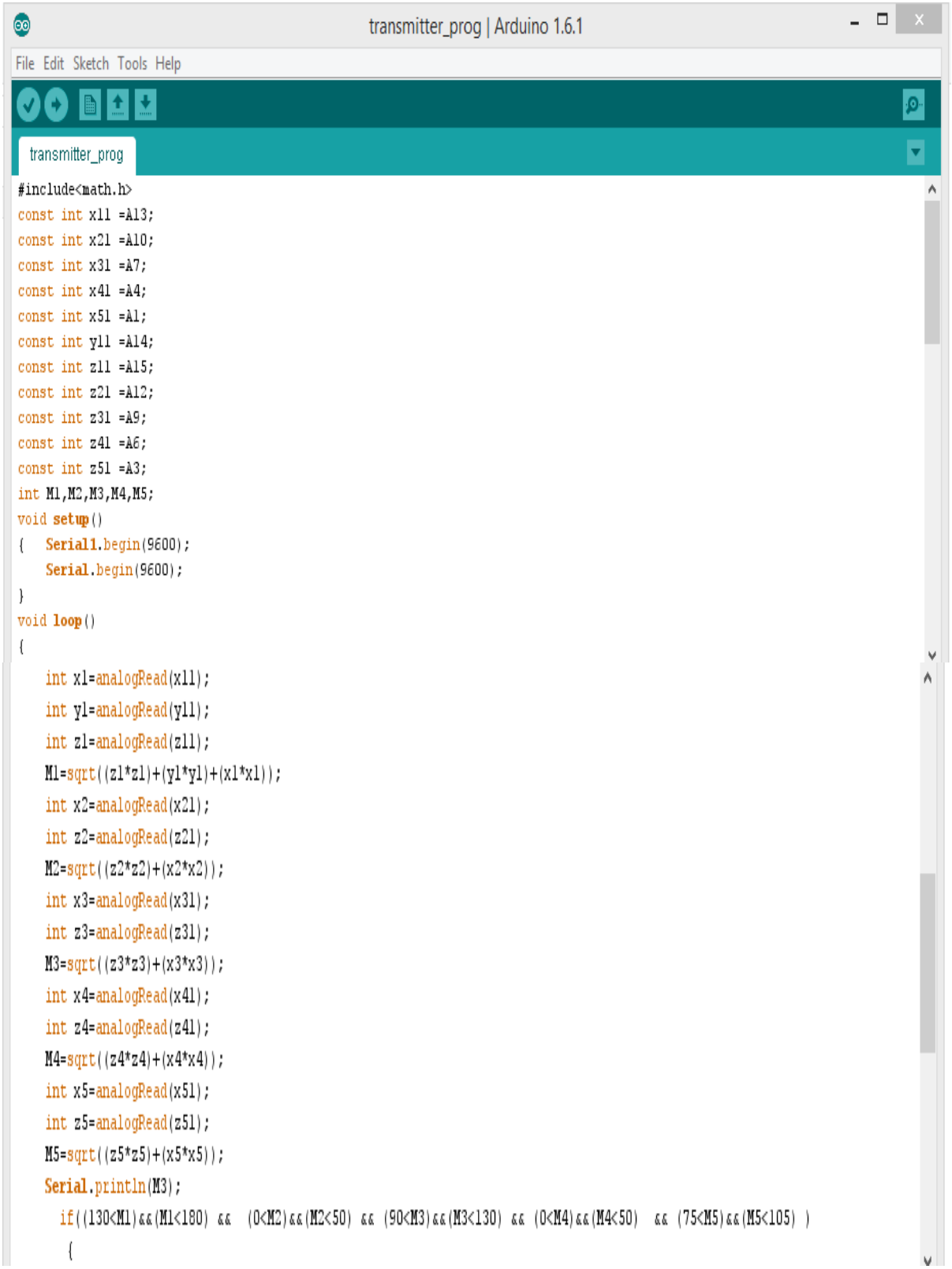


Fig 3.1: transmission circuit using flex sensor

3.1.2 Final circuit

In the final circuit accelerometers have replaced flex sensors.

3.1.2.1 Transmitter Section



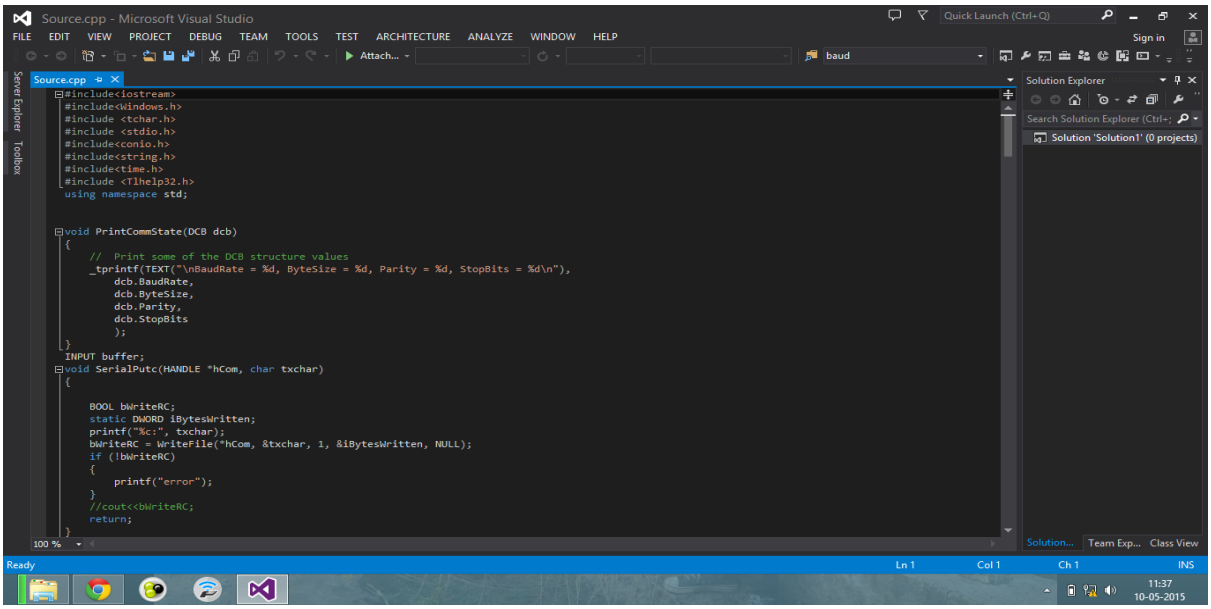

```
#include<math.h>
const int x1l =A13;
const int x2l =A10;
const int x3l =A7;
const int x4l =A4;
const int x5l =A1;
const int y1l =A14;
const int z1l =A15;
const int z2l =A12;
const int z3l =A9;
const int z4l =A6;
const int z5l =A3;
int M1,M2,M3,M4,M5;
void setup()
{   Serial1.begin(9600);
    Serial.begin(9600);
}
void loop()
{
    int x1=analogRead(x1l);
    int y1=analogRead(y1l);
    int z1=analogRead(z1l);
    M1=sqrt((z1*z1)+(y1*y1)+(x1*x1));
    int x2=analogRead(x2l);
    int z2=analogRead(z2l);
    M2=sqrt((z2*z2)+(x2*x2));
    int x3=analogRead(x3l);
    int z3=analogRead(z3l);
    M3=sqrt((z3*z3)+(x3*x3));
    int x4=analogRead(x4l);
    int z4=analogRead(z4l);
    M4=sqrt((z4*z4)+(x4*x4));
    int x5=analogRead(x5l);
    int z5=analogRead(z5l);
    M5=sqrt((z5*z5)+(x5*x5));
    Serial.println(M3);
      if((130<M1)&&(M1<180) &&  (0<M2)&&(M2<50) && (90<M3)&&(M3<130) && (0<M4)&&(M4<50)  && (75<M5)&&(M5<105) )
       {
```

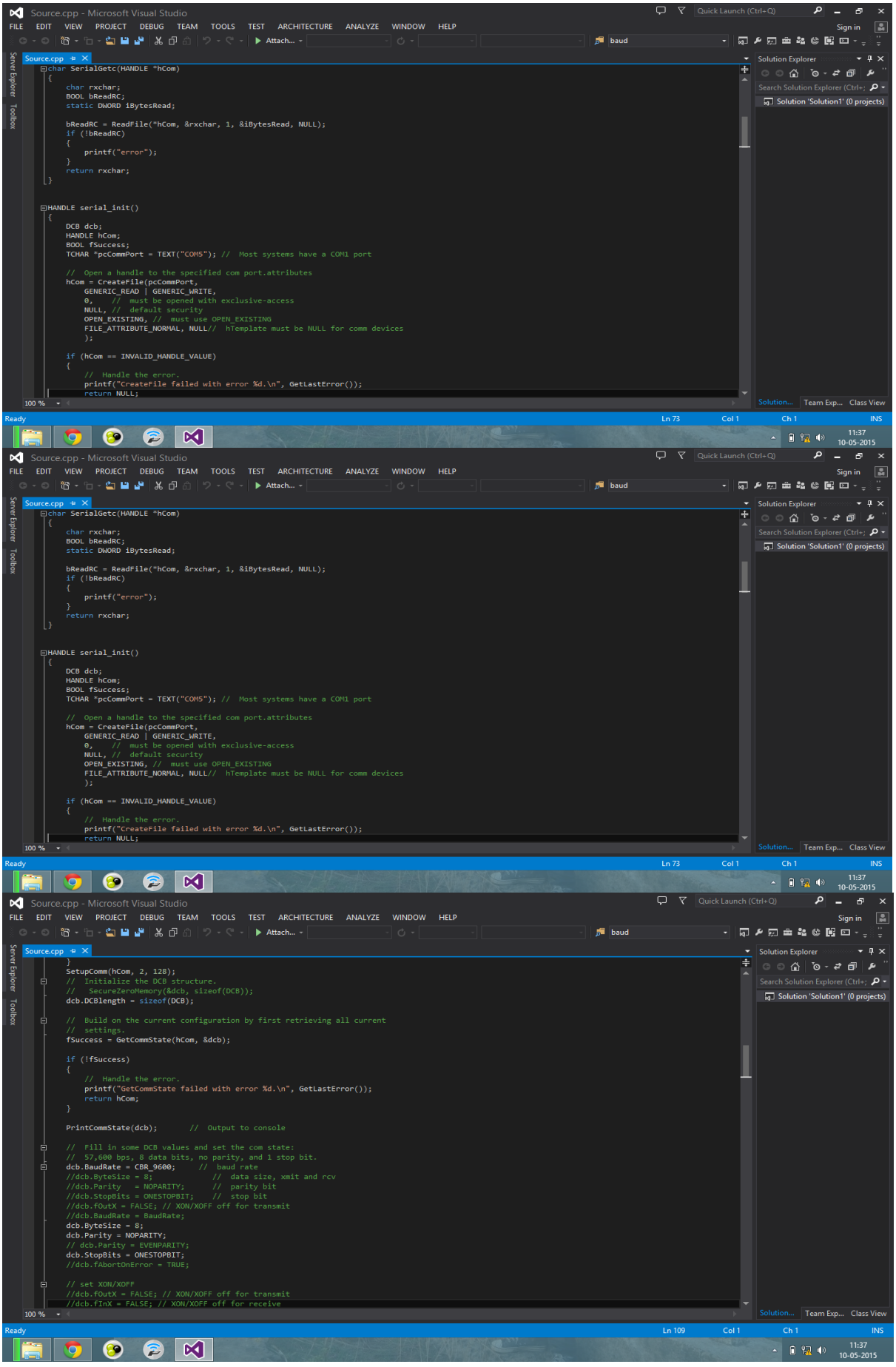
```
        Serial1.write(0);
        Serial.write(0);
      }
    else if((130<M1)&&(M1<180) &&  (110<M2)&&(M2<140) && (90<M3)&&(M3<130) &&  (0<M4)&&(M4<50)  && (75<M5)&&(M5<105) )
    {
        Serial1.write(1);
      }
    else if((130<M1)&&(M1<180) &&  (110<M2)&&(M2<140) && (130<M3)&&(M3<160) &&  (0<M4)&&(M4<50)  && (75<M5)&&(M5<105))
    {
        Serial1.write(2);
      }
    else if((0<M1)&&(M1<50) &&  (110<M2)&&(M2<140) && (100<M3)&&(M3<160) &&  (0<M4)&&(M4<50)  && (75<M5)&&(M5<105))
    {
        Serial1.write("3");
      }
    else {
        Serial1.write("4");
        Serial.print("4");
      }
}
```

1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

3.1.2.2 Bluetooth Module

Program in visual studio for transmission of data between two external bluetooths keeping internal Bluetooth as relay





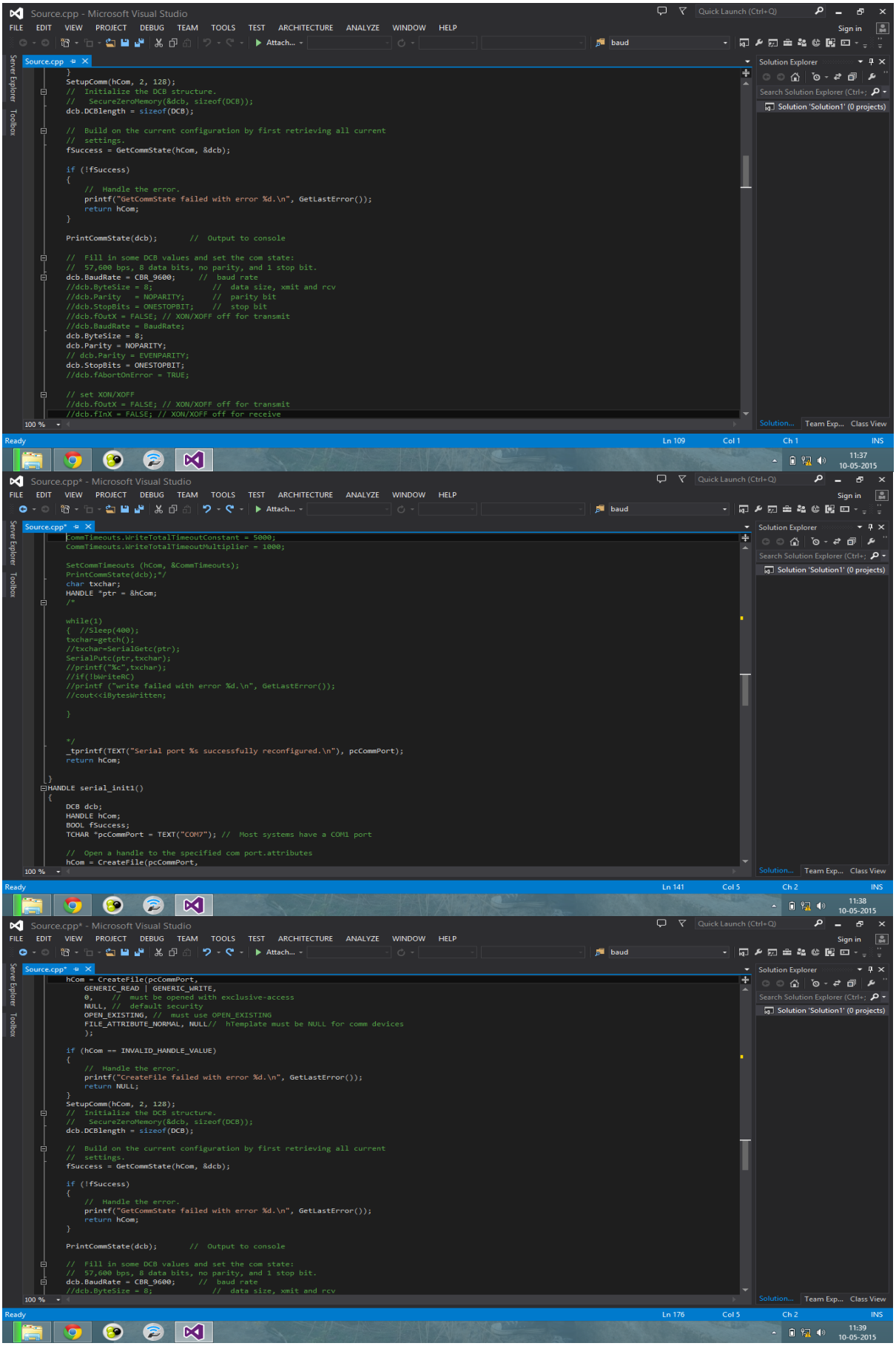

```
char SerialGetc(HANDLE *hCom)
{
    char rxchar;
    BOOL bReadRC;
    static DWORD iBytesRead;

    bReadRC = ReadFile(*hCom, &rxchar, 1, &iBytesRead, NULL);
    if (!bReadRC)
    {
        printf("error");
    }
    return rxchar;
}


HANDLE serial_init()
{
    DCB dcb;
    HANDLE hCom;
    BOOL fSuccess;
    TCHAR *pcCommPort = TEXT("COM5"); //  Most systems have a COM1 port

    //  Open a handle to the specified com port.attributes
    hCom = CreateFile(pcCommPort,
        GENERIC_READ | GENERIC_WRITE,
        0,    //  must be opened with exclusive-access
        NULL, //  default security
        OPEN_EXISTING, //  must use OPEN_EXISTING
        FILE_ATTRIBUTE_NORMAL, NULL//  hTemplate must be NULL for comm devices
        );

    if (hCom == INVALID_HANDLE_VALUE)
    {
        //  Handle the error.
        printf("CreateFile failed with error %d.\n", GetLastError());
        return NULL;
```

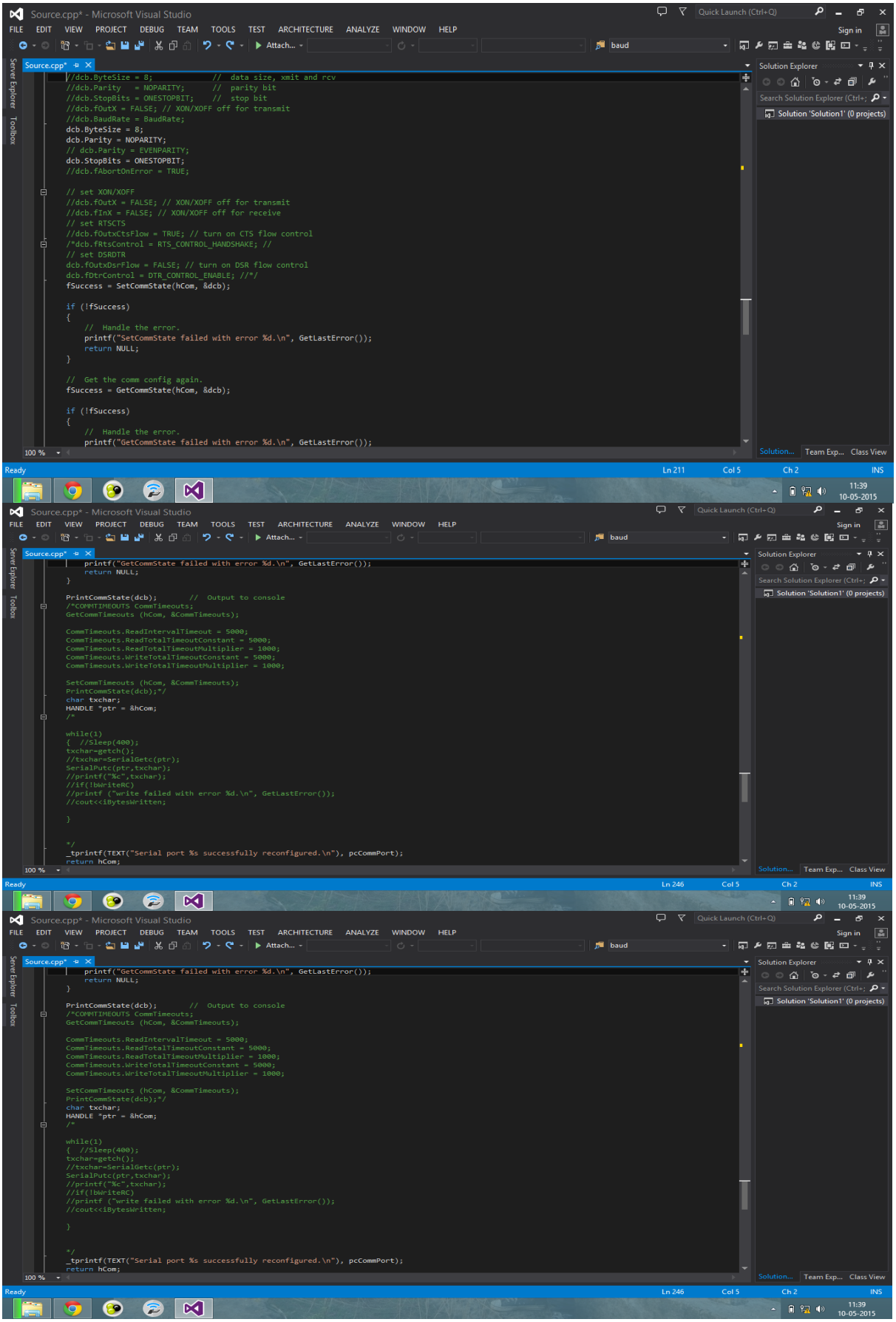


```
char SerialGetc(HANDLE *hCom)
{
    char rxchar;
    BOOL bReadRC;
    static DWORD iBytesRead;

    bReadRC = ReadFile(*hCom, &rxchar, 1, &iBytesRead, NULL);
    if (!bReadRC)
    {
        printf("error");
    }
    return rxchar;
}


HANDLE serial_init()
{
    DCB dcb;
    HANDLE hCom;
    BOOL fSuccess;
    TCHAR *pcCommPort = TEXT("COM5"); //  Most systems have a COM1 port

    //  Open a handle to the specified com port.attributes
    hCom = CreateFile(pcCommPort,
        GENERIC_READ | GENERIC_WRITE,
        0,    //  must be opened with exclusive-access
        NULL, //  default security
        OPEN_EXISTING, //  must use OPEN_EXISTING
        FILE_ATTRIBUTE_NORMAL, NULL//  hTemplate must be NULL for comm devices
        );

    if (hCom == INVALID_HANDLE_VALUE)
    {
        //  Handle the error.
        printf("CreateFile failed with error %d.\n", GetLastError());
        return NULL;
```

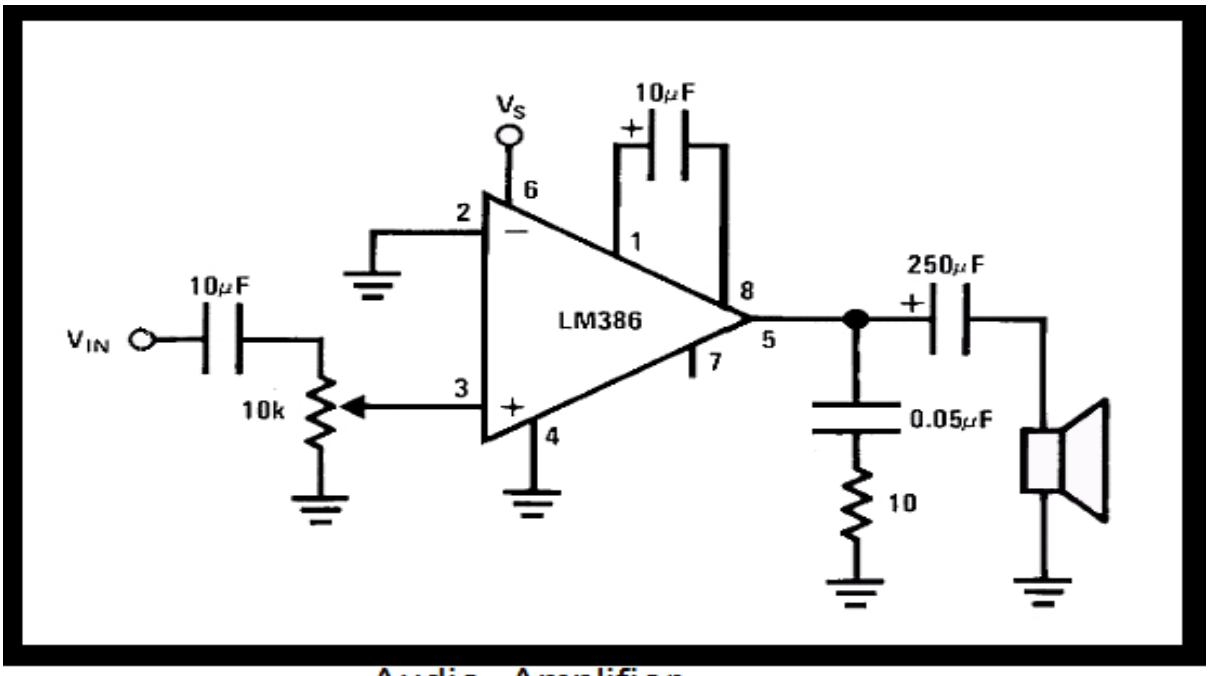


```
    }
    SetupComm(hCom, 2, 128);
    //  Initialize the DCB structure.
    //  SecureZeroMemory(&dcb, sizeof(DCB));
    dcb.DCBlength = sizeof(DCB);

    //  Build on the current configuration by first retrieving all current
    //  settings.
    fSuccess = GetCommState(hCom, &dcb);

    if (!fSuccess)
    {
        //  Handle the error.
        printf("GetCommState failed with error %d.\n", GetLastError());
        return hCom;
    }

    PrintCommState(dcb);       //  Output to console

    //  Fill in some DCB values and set the com state:
    //  57,600 bps, 8 data bits, no parity, and 1 stop bit.
    dcb.BaudRate = CBR_9600;     //  baud rate
    //dcb.ByteSize = 8;             //  data size, xmit and rcv
    //dcb.Parity   = NOPARITY;      //  parity bit
    //dcb.StopBits = ONESTOPBIT;    //  stop bit
    //dcb.fOutX = FALSE; // XON/XOFF off for transmit
    //dcb.BaudRate = BaudRate;
    dcb.ByteSize = 8;
    dcb.Parity = NOPARITY;
    // dcb.Parity = EVENPARITY;
    dcb.StopBits = ONESTOPBIT;
    //dcb.fAbortOnError = TRUE;

    // set XON/XOFF
    //dcb.fOutX = FALSE; // XON/XOFF off for transmit
    //dcb.fInX = FALSE; // XON/XOFF off for receive
```


Source.cpp - Microsoft Visual Studio
Source.cpp* - Microsoft Visual Studio
Source.cpp* - Microsoft Visual Studio
FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Solution Explorer
Ready


Source.cpp* - Microsoft Visual Studio
FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Sign in
Quick Launch (Ctrl+Q)
Attach...
baud
Source.cpp*
Solution Explorer
Search Solution Explorer (Ctrl+;
Solution 'Solution1' (0 projects)
Server Explorer
Toolbox
//dcb.ByteSize = 8; // data size, xmit and rcv
//dcb.Parity = NOPARITY; // parity bit
//dcb.StopBits = ONESTOPBIT; // stop bit
//dcb.fOutX = FALSE; // XON/XOFF off for transmit
//dcb.BaudRate = BaudRate;
dcb.ByteSize = 8;
dcb.Parity = NOPARITY;
// dcb.Parity = EVENPARITY;
dcb.StopBits = ONESTOPBIT;
//dcb.fAbortOnError = TRUE;
// set XON/XOFF
//dcb.fOutX = FALSE; // XON/XOFF off for transmit
//dcb.fInX = FALSE; // XON/XOFF off for receive
// set RTSCTS
//dcb.fOutxCtsFlow = TRUE; // turn on CTS flow control
/*dcb.fRtsControl = RTS_CONTROL_HANDSHAKE; //
// set DSRDTR
dcb.fOutxDsrFlow = FALSE; // turn on DSR flow control
dcb.fDtrControl = DTR_CONTROL_ENABLE; //*/
fSuccess = SetCommState(hCom, &dcb);
if (!fSuccess)
{
// Handle the error.
printf("SetCommState failed with error %d.\n", GetLastError());
return NULL;
}
// Get the comm config again.
fSuccess = GetCommState(hCom, &dcb);
if (!fSuccess)
{
// Handle the error.
printf("GetCommState failed with error %d.\n", GetLastError());
100 %
Solution... Team Exp... Class View
Ready
Ln 211 Col 5 Ch 2 INS
11:39
10-05-2015
printf("GetCommState failed with error %d.\n", GetLastError());
return NULL;
}
PrintCommState(dcb); // Output to console
/*COMMTIMEOUTS CommTimeouts;
GetCommTimeouts (hCom, &CommTimeouts);
CommTimeouts.ReadIntervalTimeout = 5000;
CommTimeouts.ReadTotalTimeoutConstant = 5000;
CommTimeouts.ReadTotalTimeoutMultiplier = 1000;
CommTimeouts.WriteTotalTimeoutConstant = 5000;
CommTimeouts.WriteTotalTimeoutMultiplier = 1000;
SetCommTimeouts (hCom, &CommTimeouts);
PrintCommState(dcb);*/
char txchar;
HANDLE *ptr = &hCom;
/*
while(1)
{ //Sleep(400);
txchar=getch();
//txchar=SerialGetc(ptr);
SerialPutc(ptr,txchar);
//printf("%c",txchar);
//if(!bWriteRC)
//printf ("write failed with error %d.\n", GetLastError());
//cout<<iBytesWritten;
}
*/
_tprintf(TEXT("Serial port %s successfully reconfigured.\n"), pcCommPort);
return hCom;
Ready
Ln 246 Col 5 Ch 2 INS
11:39
10-05-2015

3.1.2.3 Receiver Section



Audio Amplifier

Fig 3.2 Audio Amplifier

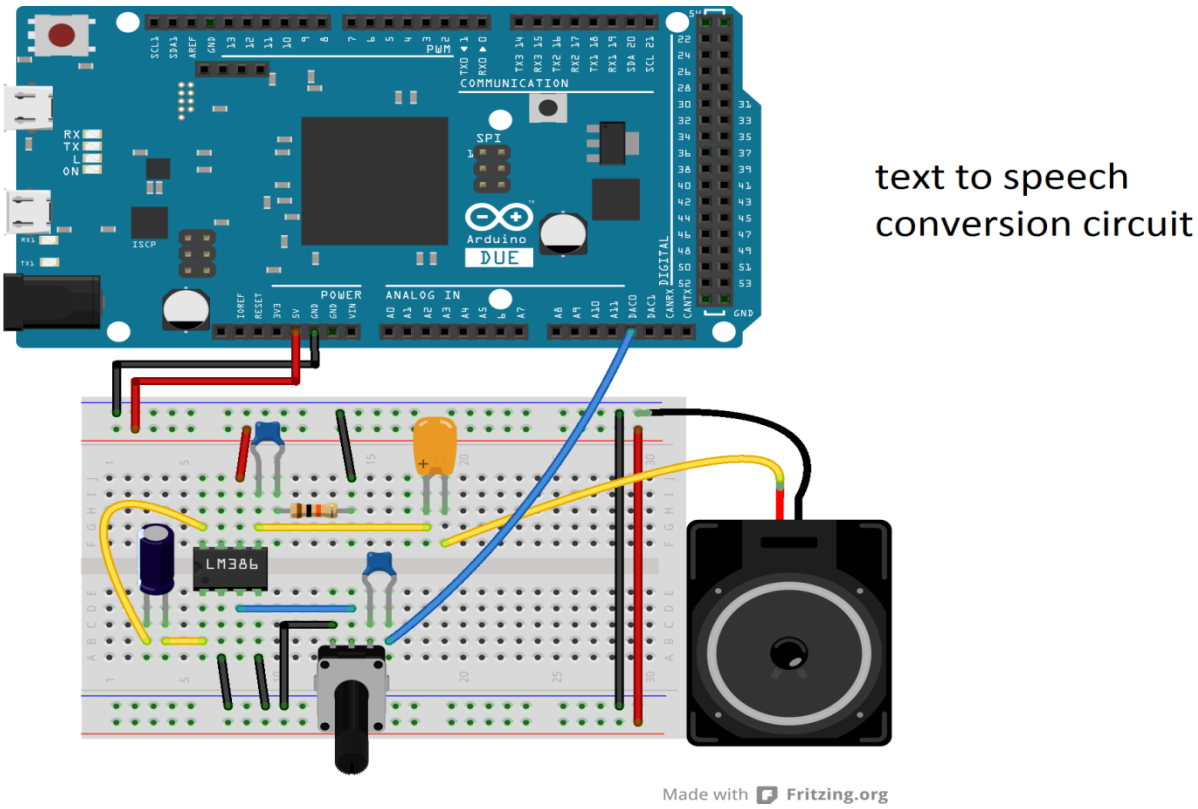
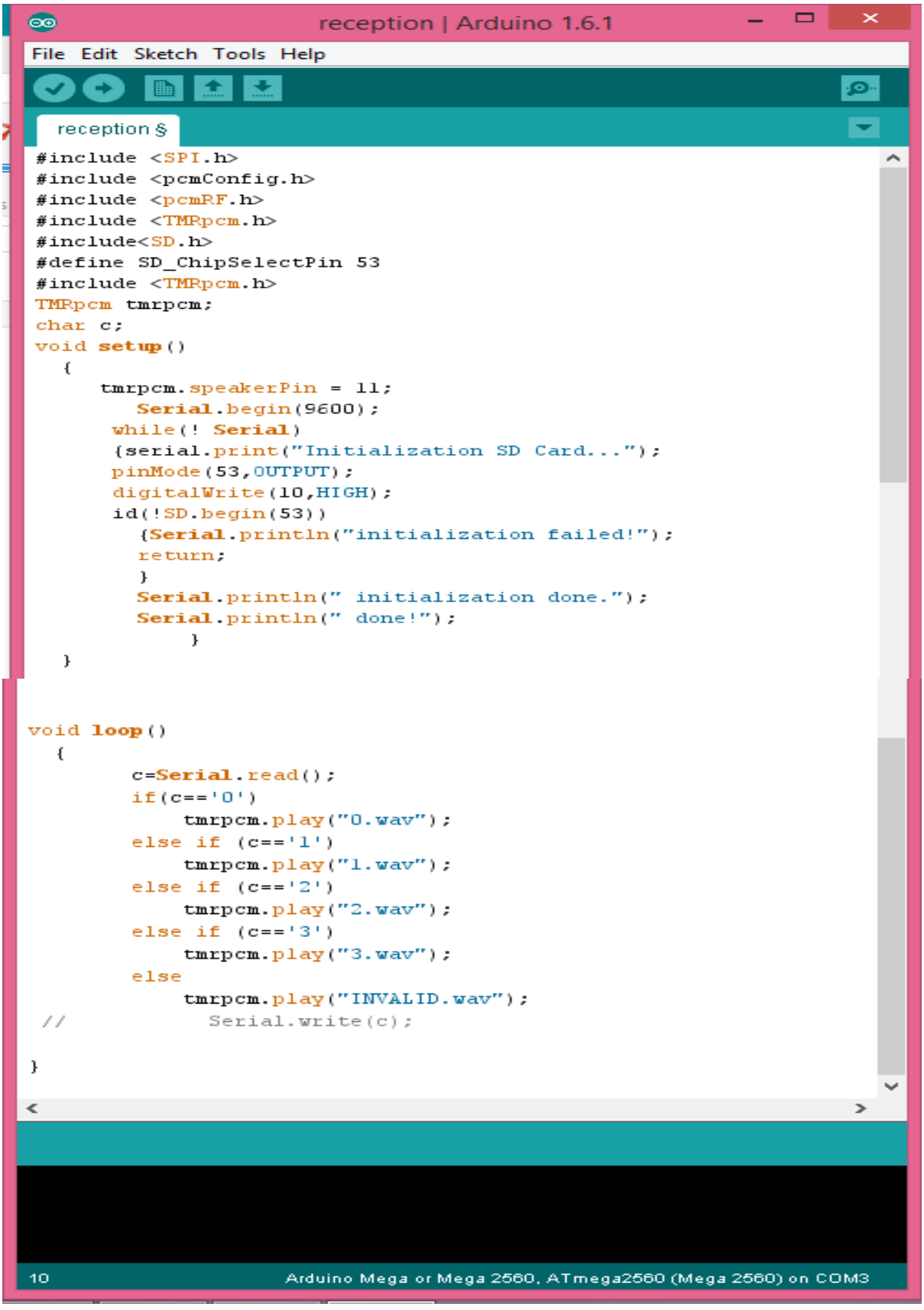


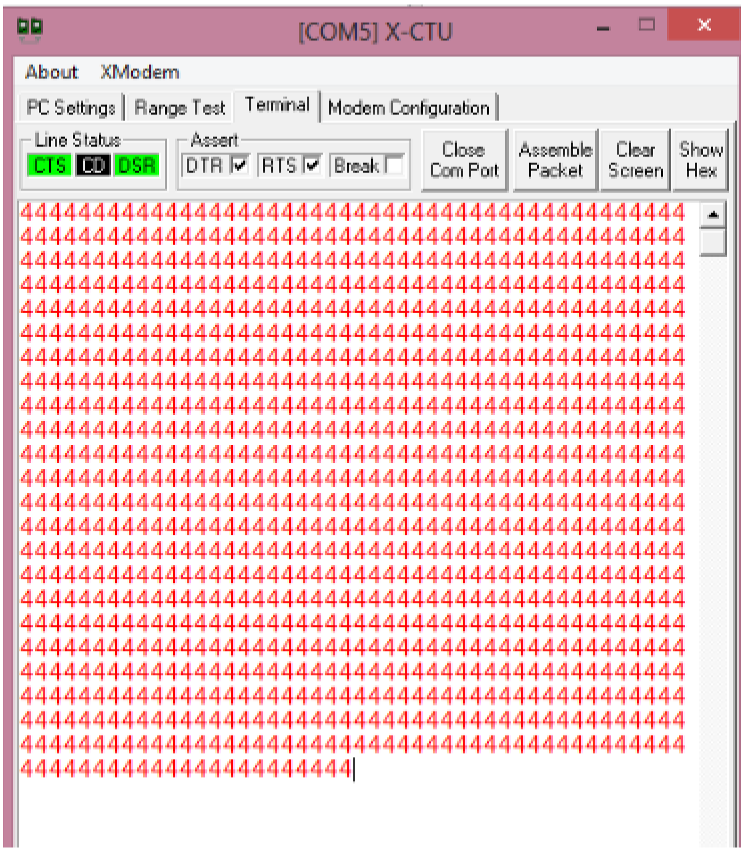
Fig 3.3 Text to speech conversion circuit

Code for playing audio file from Arduino:

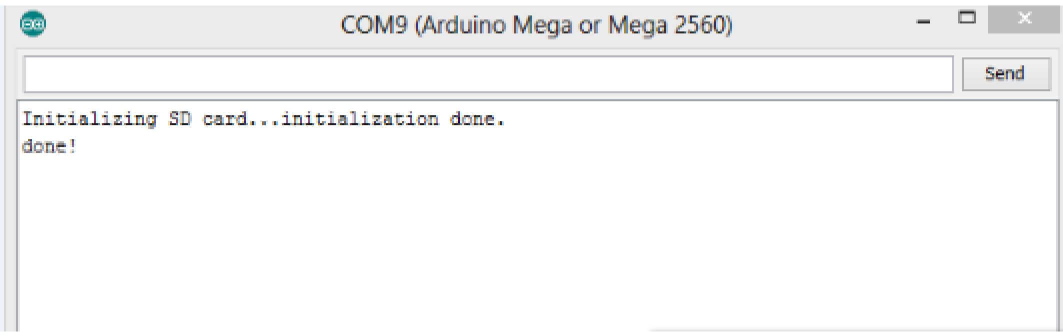


Output:

Checking transmission through X-CTU



Checking initialization at reception side



3.2 Discussion

In the project we have first used a set of five flex sensors. But they could only measure the bending in a particular direction i.e. there were variations in actual characters but there was no variation in transmitted data.

Secondly, we used a set of four flex sensors and one accelerometer. But frequent use of the flex sensors led to their damage because of excessive bending.

Finally a set of five accelerometers were used. It has one disadvantage that is we have to change the range of data for each particular finger for every number depending on the speed of hand movements of the person.

.The hand gesture for a particular number provides the required data (i.e. acceleration measured by accelerometer). This data is fed into Arduino Mega 2560. Here the data from all the five fingers is mapped into the number represented by the particular hand gesture. This number is then send from Bluetooth device hc-05 to internal Bluetooth i.e. hc-06. This internal Bluetooth acts as a relay to connect two external Bluetooth devices, the other one to receive the data.

The received data is passed to Arduino mega. Few wav files are stored in SD card. Respective wav file is played in accordance with the data .the sound is connected to the speaker or a headphone.

C H A P T E R – 4
FUTURE SCOPE AND CONCLUSION

4.1 Future scope

The project involves distinguishing among five different alphabets of English language. Future work may include recognition of all the English alphabets. Further, we may move on to recognizing words, from as large a dictionary as possible, from Indian Sign Language. Another method to improve the performance is by using android programming to use google to receive data and decode the received alphanumeric character.

4.2 Conclusion

- Talking gloves was implemented for four numbers (0 , 1, 2 ,3)
- The system was tested for ten different inputs per number.
- The .wav files were played corresponding to the respective numbers
- The sound is not audible properly due to SD card problem

REFERENCES

[1] Auti, Abhijit; Puranik, V.G.; Kureshi, A.K. (2014) "Speaking Gloves for Speechless Persons." International Journal of Innovative Research in Science, Engineering and Technology. Vishwabharati Academy's College of Engineering.

[2] Havalagi, Praveenkumar S; Nivedita (2013). "The Amazing Digital Gloves That Give Voice to the Voiceless." International Journal of Advances in Engineering & Technology. ISSN: 2231-1963.

[3] Khan, Yasir Niaz; Mehdi, Syed Atif (2009). "Sign Language Recognition using Sensor Gloves." National University of Computer and Engineering Sciences, Lahore.

[4] Karlan, G.R. (1990). "Manual Communication: Implications for Education." Gallaudet University Press, Washington, DC, USA. P 51 – 185.